ELISTdbinst.8100.Final.SOL7.SVD

Defense Information Infrastructure (DII)
Common Operating Environment (COE)

Software Version Description (SVD)
for the
Enhanced Logistics Intratheater Support Tool (ELIST)
Database Instance Segment Version 8.1.0.0

for Solaris 7

26 February 2002

Prepared for:

Military Traffic Management Command
Transportation Engineering Agency
720 Thimble Shoals Boulevard
Newport News, VA 23606

Prepared by:

Argonne National Laboratory
Decision and Information Sciences Division
9700 South Cass Avenue
Argonne, IL 60439

Table of Contents

List of Tables

1. Scope

This document is the *Software Version Description (SVD) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment*. It contains basic information about the segment.

1.1 Identification

The ELIST Database Instance Segment is one of seven segments that make up the DII COE ELIST mission application. The distributed segment software, data, and documentation are **Unclassified**.

Table 1 identifies all the segments of the ELIST mission application. In the table, each segment is given a number by which it may be referenced in this document. The table also gives the name, the segment type (and, if a data segment, the segment scope), the current version number, and the directory name assigned to each segment.

Table 1. Segments of the ELIST Mission Application

Segment Number	Segment Name	Segment Type / Scope	Version Number	Directory Name
1	ELIST Global Data Segment	Data / Global	8.1.0.0	`ELISTglob`
2	ELIST Database Instance Segment	Data / Segment	8.1.0.0	`ELISTdbinst`
3	ELIST Database Fill Segment	Data / Local	8.1.0.0	`ELISTdbfill`
4	ELIST Database Segment	Database	8.1.0.0	`ELISTdb`
5	ELIST Database Utility Segment	Software	8.1.0.0	`ELISTdbutil`
6	ELIST Software Segment	Software	8.1.0.0	`ELISTexec`
7	ELIST Reference Data Segment	Data / Local	8.1.0.0	`ELISTrefdata`

All seven segments have the following identification properties in common:

Segment Prefix[1]: ELIST

Platform(s)[2]: Sun/Solaris 7

DII COE Versions: 4.2.0.0P4 or later

All seven of the ELIST segments must be installed before you can use the ELIST mission application.[3]

Refer to the *Introduction to the Enhanced Logistics Intratheater Support Tool (ELIST) Mission Application and its Segments: Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment* for the following:

[1] Note carefully that all segments have the same prefix. This is not typical of multisegment DII COE mission applications.

[2] Implementation of the ELIST segments for PC/Windows NT 4.0 will follow shortly. This documentation covers only the Sun/Solaris 7 platform but will be supplemented or replaced when an implementation becomes available for NT.

[3] To save space, however, the ELIST Database Fill Segment can be removed after successfully installing the ELIST Database Segment.

- an overview of the mission application and all of its segments in the context of the application;

- considerations applicable to the use of ELIST with classified data;

- the definitions of key concepts and terms used throughout the ELIST documentation;

- a complete list of the available ELIST documentation;

- a brief history of ELIST; and

- basic information pertinent to the client/server configuration and installation of the ELIST segments.

1.2 Segment Overview

The purpose of the ELIST Database Instance Segment is to ensure the existence of a database instance in which the ELIST database can be created. The actual creation of the ELIST database in that database instance is accomplished by the subsequent installation of a different segment, called the ELIST Database Segment.

Because the database administrator (DBA) at some sites may prefer to accommodate the databases required by several separate applications within a single shared database instance, while another DBA may prefer to associate those separate databases with unique database instances, the ELIST Database Instance Segment gives the installer two choices: using an existing database instance or creating a new database instance. The capability to designate an existing database instance also allows an experienced DBA to create a new database instance for ELIST by independent means. Such a capability might be used, for example, to establish a database instance for ELIST with properties that differ from those that would be assigned by the ELIST Database Instance Segment, were it allowed to create the instance.

The ELIST Database Instance Segment should be installed by, or in the presence of, a DBA (using the usual `sysadmin` account). If the option of using an existing database instance is chosen, the installer is responsible for ensuring that the properties of the existing database instance are suitable for ELIST. On the other hand, if the option of creating a new database instance is chosen, the installer is asked to select a `SYSTEM` password for the new instance, which the segment then creates with the appropriate properties. Those properties are outlined in the *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment*.

The *System Administrator's Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment* provides guidance for recovering from errors that can occur during the segment's installation or deinstallation. It also outlines methods for creating an ELIST database instance before installing the segment, in case the installer chooses to use an existing instance rather than allowing the segment to create a new instance when it is installed. Finally, it contains instructions for changing the password of the `SYSTEM` account of the database instance, should that become necessary after the segment is installed.

1.3 Points of Contact

The development of ELIST was sponsored by the Military Traffic Management Command (MTMC) Transportation Engineering Agency (TEA). ELIST was implemented by MTMCTEA and Argonne National Laboratory (ANL). For administrative and program management support, including the permission that DISA Configuration Management (CM) requires to release the distribution media for the ELIST mission application to prospective users, contact

Doug Barbour, ELIST Project Officer
MTMCTEA
720 Thimble Shoals Blvd.
Suite 130
Newport News, VA
23606-0276
Tel. (757) 599-1663
Fax (757) 599-1564
barbourd@tea-emh1.army.mil

For technical support of a segment-specific (*i.e.*, DII COE-related) nature, contact

Kenneth Dritz, DII COE Integrator
Decision and Information Sciences Division
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439-4844
Tel. (630) 252-7217
Fax (630) 252-5128
dritz@anl.gov

For technical support of a more general nature (*i.e.*, related to issues of functionality), contact

Charles VanGroningen, ELIST Project Manager
Decision and Information Sciences Division
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439-4844
Tel. (630) 252-5308
Fax (630) 252-6073
vang@anl.gov

1.4 Product Information

Product Qualification: Completed in-house testing

Product Restrictions: Releasable to military organizations (like OSD; Joint Staff; EUCOM, CENTCOM, and other CINCs) and perhaps other government organizations with approval of MTMCTEA. Consult Chapter 8 of Army Regulation 5-11 (see citation in Section 2.1.4, below), then contact the ELIST Project Officer at the address given in Section 1.3

Product Dependencies: ORACLE DataBase Instance (ORADBI) 2.1.0.0/8.1.6 or later
ELIST Global Data Segment (ELIST) 8.1.0.0 or later

2. Referenced Documents

The following other documents are referenced in this document.

2.1 Government Documents

2.1.1 DII COE ELIST Documents

Introduction to the Enhanced Logistics Intratheater Support Tool (ELIST) Mission Application and its Segments: Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7, ELIST.8100.Draft.SOL7.Intro, Argonne National Laboratory, 26 February 2002

Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7, ELIST.8100.Draft.SOL7.IP, Argonne National Laboratory, 26 February 2002

System Administrator's Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment Version 8.1.0.0 for Solaris 7, ELISTdbinst.8100.Draft.SOL7.SAM, Argonne National Laboratory, 26 February 2002

Software Test Plan/Description/Report (STP/STD/STR) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7, ELIST.8100.Draft.SOL7.STP_STD_STR, Argonne National Laboratory, 26 February 2002

2.1.2 Other DII COE Documents

N/A.

2.1.3 Other ELIST Documents

DII COE Integration and Runtime Specification, Version 4.1, DISA/JIEO, 3 October 2000

2.1.4 Other Government Documents

Army Regulation 5-11, *Management of Army Models and Simulations*, Army HQ, Washington DC, 10 July 1997. Available on the World Wide Web at http://www.amso.army.mil/ar5-11. POC: Debra Ridgeway, Army Model Simulation Office, DSN 329-0012 ext. 24, comm. (703) 601-0012 ext. 24, email debra.ridgeway@hqda.army.mil

2.2 Non-Government Documents

N/A.

This page intentionally left blank.

3. Version Description

3.1 Inventory of Materials Released

NOTE: See Section 1.4 for Product (*i.e.*, Releasability) Restrictions.

The ELIST Database Instance Segment for Solaris 7, Version 8.1.0.0, CM Number 10102, 26 February 2002, is distributed in `MakeInstall` format on two CDs, one externally labeled "Master" and the other "Backup," and both externally labeled "ELIST Mission Application Software." The other segments of the ELIST mission application are distributed on the same CDs.

Documentation for the ELIST Database Instance Segment is also distributed on two CDs, one externally labeled "Master" and the other "Backup," and both externally labeled "ELIST Mission Application Documentation." Documentation for the other segments of the ELIST mission application is distributed on the same CDs. On these CDs, the documentation pertinent to the ELIST Database Instance Segment can be found in the following directories:

- `ELIST` directory
  - *Introduction to the Enhanced Logistics Intratheater Support Tool (ELIST) Mission Application and its Segments: Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7*, ELIST.8100.Draft.SOL7.Intro, Argonne National Laboratory, 26 February 2002
  - *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment Version 8.1.0.0, Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7*, ELIST.8100.Draft.SOL7.IP, Argonne National Laboratory, 26 February 2002
  - *Software Test Plan/Description/Report (STP/STD/STR) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment* Version 8.1.0.0, *Database Instance Segment Version 8.1.0.0, Database Fill Segment Version 8.1.0.0, Database Segment Version 8.1.0.0, Database Utility Segment Version 8.1.0.0, Software Segment Version 8.1.0.0, and Reference Data Segment Version 8.1.0.0 for Solaris 7*, ELIST.8100.Draft.SOL7.STP_STD_STR, Argonne National Laboratory, 26 February 2002
- `ELISTdbinst` directory
  - *Software Version Description (SVD) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment Version*

8.1.0.0 for Solaris 7, ELISTdbinst.8100.Draft.SOL7.SVD, Argonne National Laboratory, 26 February 2002

– *System Administrator Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment Version 8.1.0.0 for Solaris 7*, ELISTdbinst.8100.Draft.SOL7.SAM, Argonne National Laboratory, 26 February 2002

3.2 Pertinent Documentation

All pertinent documentation is delivered with the segment (see Section 3.1).

3.3 Inventory of Software Contents

The directory structure of the ELIST Database Instance Segment, as distributed, can be inferred from the listing of the `FileAttribs` descriptor file in Appendix A.7.

The segment is a data segment of segment scope. It contains directories and data files, some of which are used by the ORACLE DataBase Administration (ORADBI) segment. There are no executables and no scripts except for the installation and deinstallation scripts, including various database instance creation scripts (*e.g.*, SQL scripts).

3.4 Changes Installed

N/A.

3.5 Waivers

See Appendix C.

3.6 Adaptation Data

Although an effort was made in developing the ELIST Database Instance Segment to determine parameters that would be suitable for an ELIST database instance at a wide variety of sites, some sites may have unique requirements. At such sites, the DBA can satisfy those requirements by creating the ELIST database instance by means apart from this segment, before the segment is installed; then, responding to a prompt when the segment is installed, the DBA will indicate that the segment should use the instance that already exists. Ways to create the instance in this case are discussed in the *System Administrator Manual (SAM) for the Enhanced Logistics Intratheater Support Tool (ELIST) Database Instance Segment*.

3.7 Installation Instructions

Installation instructions for all the segments of the ELIST mission application can be found in the *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment*.

The computer system resources required by the ELIST Database Instance Segment are as follows:

- RAM required: 20 MB[4]
- Disk space required: 79 KB

The figure given above for disk space required is for storing the segment directories and files. The total initial RDBMS disk space required for an ELIST database instance in the relevant Oracle directories is approximately 1.327 GB. (See the *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment* for further information.)

There are no "features" of the ELIST Database Instance Segment that can be executed after installation; all the processing involved in the creation of a database instance takes place during installation, under the control of the `PostInstall` script. Scripts for creating and managing ELIST database user accounts are provided as features of the ELIST Database Utility Segment.

3.8 Possible Problems and Known Errors

There were no known problems or errors when this document was prepared.

[4] Nominal requirement for the kernel and RDBMS tools invoked by the scripts of this segment.

This page intentionally left blank.

4. Notes

The ELIST Database Instance Segment must be installed on a platform configured as a database server. In the current implementation, that platform must be a Sun workstation. It is not necessary to have created any previous database instance on that platform. Installation of the ELIST Database Instance Segment requires that both the ORACLE DataBase Instance (ORADBI) segment and the ELIST Global Data Segment (ELIST) be resident on the same platform. The prerequisites for those segments, in turn, are documented in their SVDs. Further information on system configuration can be found in the *Installation Procedures (IP) for the Enhanced Logistics Intratheater Support Tool (ELIST) Global Data Segment, Database Instance Segment, Database Fill Segment, Database Segment, Database Utility Segment, Software Segment, and Reference Data Segment*.

Several files in this segment necessarily have permissions that are less restrictive than permitted by Table 4-1 in the *DII COE Integration and Runtime Specification* (I&RTS). In particular, several files underneath this segment's `data` directory have execute permission (for both the user and the group), which is prohibited by Table 4-1. This segment's use of the ORACLE DataBase Instance (ORADBI) segment *requires* certain database instance creation scripts to be present within the `data` directory, and they *must* have execute permission. In accordance with I&RTS Appendix B checklist item 6-5, Chief Engineer permission has been requested in connection with the execute permission given to these files. Item 6-5 also requires the affected files to be listed in the SVD. The files in question are as follows:

```
/h/ELISTdbinst/data/dbdesc/create/Create_SID
/h/ELISTdbinst/data/dbdesc/create/SIDAlterTablespace.sh
/h/ELISTdbinst/data/dbdesc/create/SIDjava.sh
/h/ELISTdbinst/data/dbdesc/create/SIDrun.sh
/h/ELISTdbinst/data/dbdesc/create/SIDrun1.sh
/h/ELISTdbinst/data/dbdesc/create/SIDrun2.sh
/h/ELISTdbinst/data/dbdesc/create/SIDsqlplus.sh
```

The following acronyms are (or may be) used in this document.

Acronym	Definition
ANL	Argonne National Laboratory
CD	Compact Disk
CENTCOM	Central Command
CINC	Combined Intelligence Center (as used in this document)
CM	Configuration Management
COE	Common Operating Environment
COTS	Commercial Off-the-Shelf
DB	Database
DBA	Database Administrator
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DSN	Defense Switched Network
ELIST	Enhanced Logistics Intratheater Support Tool (DII COE segment prefix)
ETEdit	ETPFDD Editor

ETPFDD	Expanded Time Phased Force Deployment Data
EUCOM	European Command
GB	Gigabyte(s)
HQ	Headquarters
IP	Installation Procedures
I&RTS	Integration and Runtime Specification
LAN	Local Area Network
KB	Kilobyte(s)
MB	Megabyte(s)
MTMC	Military Traffic Management Command
N/A	Not Applicable
NFS	Network File System
NT	New Technology (an Operating System for Microsoft Windows)
ORADBI	ORACLE DataBase Instance (DII COE segment prefix)
OSD	Office of the Secretary of Defense
PC	Personal Computer
POC	Point of Contact
RDBMS	Relational Database Management System
RAM	Random Access Memory
SAM	System Administrator’s Manual
SQL	Structured Query Language
STD	Software Test Description
STP	Software Test Plan
STR	Software Test Report
SVD	Software Version Description
TEA	Transportation Engineering Agency

5. Acknowledgements

Argonne National Laboratory is a Federally Funded Research and Development Center operated by The University of Chicago under contract W-31-109-ENG for the United States Department of Energy.

The development of ELIST and the preparation of this document were supported by funding from the Military Traffic Management Command Transportation Engineering Agency of the United States Army.

This page intentionally left blank

6. Documentation Improvement and Feedback

Comments and other feedback on this document should be directed to:

Phone: (630) 252-7217
Fax: (630) 252-5128
Email: dritz@anl.gov

This page intentionally left blank.

Appendix A. Contents of the Descriptor Files in the `SegDescrip` Directory

The contents of the descriptor files in the `SegDescrip` directory are reproduced in this appendix.

A.1 `SegName` File

```
#===================================================================
# SegName descriptor file for the ELIST Database Instance Segment
# Segment Prefix: ELIST
# Segment Directory: ELISTdbinst
#===================================================================

$TYPE:DATA
$NAME:ELIST Database Instance Segment
$PREFIX:ELIST
$SEGMENT:ORACLE DataBase Instance:ORADBI:/h/COTS/ORADBI

# END OF FILE
```

A.2 `SegInfo` File

```
# This is the SegInfo descriptor for the ELIST Database Instance Segment.

[Data]
$SEGMENT:ORACLE DataBase Instance:ORADBI:/h/COTS/ORADBI

[Security]
UNCLASS

[Hardware]
$CPU:SPARC
$DISK:79:1
$OPSYS:SOL
$MEMORY:0

[Requires]
ORACLE DataBase Instance:ORADBI:/h/COTS/ORADBI:2.1.0.0/8.1.6
ELIST Global Data Segment:ELIST:/h/ELISTglob:8.1.0.0

[Direct]
$ROOT:PostInstall
$ROOT:DEINSTALL

[FilesList]
$PATH:/h/data/global/ELISTglob/data
$FILES
  # This segment writes into the following file, which is subsequently read
  # by other segments of the ELIST mission application:
  Instance_Info
$PATH:/h/COTS/ORADBI
$FILES
  # This segment directly invokes the following public APIs of ORADBI:
  bin/ORADBI_getSids
  bin/ORADBI_auto
  bin/ORADBI_register
$DIRS
  # Through a call to ORADBI_register in this segment's PostInstall, the link:
  data/dbdesc/ELISTdbinst
  # is created, pointing to:
  data/ELISTdbinst/data/dbdesc
  # The link is removed by the same API, called from this segment's DEINSTALL.
  # Facilities of ORADBI used by this segment write a log and other files
  # in the following directory:
  data/db
```

```
# The following Oracle files and directories are those used or modified by
# ORADBI_auto, as reflected in the [FilesList] descriptor of ORADBI, plus a
# few others not mentioned there:
$PATH:/var/opt/oracle
$FILES
  oratab
  listener.ora
  tnsnames.ora
$PATH:/etc
$FILES
  init.d/dbora
  rc2.d/S99dbora
  rc0.d/K01dbora
$PATH:/
$DIRS
  ora01
  ora02
  ora03
  ora04
  ora01/oradata
  ora02/oradata
  ora03/oradata
  ora04/oradata
  ora01/app/oracle/admin
  ora01/app/oracle/oradba
  ora01/app/oracle/product/8.1.6/dbs
  ora01/app/oracle/product/8.1.6/rdbms/log

# END OF FILE
```

A.3 VERSION File

```
# This is the VERSION descriptor for the ELIST Database Instance Segment.
8.1.0.0 : 01/07/2002 : 10:41

# END OF FILE
```

A.4 ReleaseNotes File

```
This is the ReleaseNotes file for the ELIST Database Instance Segment.

The primary purpose of this segment is to ensure that a database instance
suitable for the ELIST mission application exists.  (A different segment, the
ELIST Database Segment, will create the actual application-oriented tablespaces
and schema objects of the ELIST database in this database instance.)

This segment must be installed on the database server platform on which the
ELIST Global Data Segment has already been installed and on which the ELIST
Database Fill Segment, the ELIST Database Segment, and the ELIST Database
Utility Segment will subsequently be installed.  That machine is the one on
which the Oracle RDBMS (ORAS) has already been installed.

This segment is designed to give the installer, assumed to be a DBA, a choice
between using an already existing database instance for ELIST or creating a new
database instance for ELIST.  In the former case, it is the installer's
responsibility to ensure that the existing database instance has the properties
that ELIST requires (see below).  In the latter case, this segment creates the
new database instance with the required properties.

If the installer chooses to use an existing database instance, he or she should
be prepared to enter the name of the instance (i.e., the SID), when prompted.
If the installer chooses to create a new database instance, he or she should be
prepared to supply the name of the new instance and to supply a password to be
assigned to the SYSTEM user in the newly created instance, when prompted.  When
the ELIST Database Segment is subsequently installed, the installer will be
prompted for the SYSTEM password of the database instance, regardless of
whether the instance was created by this segment.  Similarly, a user who
executes any of the executable features of the ELIST Database Utility Segment
(for example, to create ELIST database user accounts) will be prompted for the
SYSTEM password of the database instance, regardless of whether the instance
was created by this segment.  (In the latter case, the user also has to be an
```

administrative ELIST user, i.e., a user in the "elistadm" group, to whom the "Administrative ELIST User" profile has been assigned.)

The following script files from the data/dbdesc directory show the properties explicitly given to a new ELIST database instance. If an existing instance is to be used, the installer must ensure that it has at least these properties; if the instance is shared with other applications, properties like sizes should accommodate those other applications and ELIST together. The most unique requirement that needs to be satisfied is the provision of a large, nonstandard tablespace named BIGROLL, containing a rollback segment named BIG_RBS. Further information may be found in this segment's System Administrator's Manual (SAM).

In these scripts, certain parameters are symbolic names for which substitutions are made during the installation of the segment, as follows:

```
<ORACLE_SID>: (the instance name chosen by the installer)
<ORACLE_MNT1>: /ora01
<ORACLE_MNT2>: /ora02
<ORACLE_MNT3>: /ora03
<ORACLE_MNT4>: /ora04
<ORACLE_BASE>: /ora01/app/oracle
<ORACLE_HOME>: <ORACLE_BASE>/product/8.1.6
```

```
========================================
== File data/dbdesc/pfile/initSID.ora ==
========================================

#
# Copyright (c) 1991, 1998 by Oracle Corporation
#
##############################################################################
#
# File modified by K. Dritz, ANL
#
##############################################################################
# Example INIT.ORA file
#
# This file is provided by Oracle Corporation to help you customize
# your RDBMS installation for your site.  Important system parameters
# are discussed, and example settings given.
#
# Some parameter settings are generic to any size installation.
# For parameters that require different values in different size
# installations, three scenarios have been provided: SMALL, MEDIUM
# and LARGE.  Any parameter that needs to be tuned according to
# installation size will have three settings, each one commented
# according to installation size.
#
# Use the following table to approximate the SGA size needed for the
# three scenarious provided in this file:
#
#                     -------Installation/Database Size------
#                      SMALL           MEDIUM           LARGE
#  Block         2K    4500K            6800K           17000K
#  Size          4K    5500K            8800K           21000K
#
# To set up a database that multiple instances will be using, place
# all instance-specific parameters in one file, and then have all
# of these files point to a master file using the IFILE command.
# This way, when you change a public
# parameter, it will automatically change on all instances.  This is
# necessary, since all instances must run with the same value for many
# parameters. For example, if you choose to use private rollback' segments,
# these must be specified in different files, but since all gc_*
# parameters must be the same on all instances, they should be in one file.
#
# INSTRUCTIONS: Edit this file and the other INIT files it calls for
# your site, either by using the values provided here or by providing
# your own.  Then place an IFILE= line into each instance-specific
# INIT file that points at this file.
#
# NOTE: Parameter values suggested in this file are based on conservative
# estimates for computer memory availability. You should adjust values upward
# for modern machines.
```

```
#
###############################################################################

db_name = "<ORACLE_SID>"
instance_name = <ORACLE_SID>

service_names = <ORACLE_SID>


control_files = ( "<ORACLE_MNT2>/oradata/<ORACLE_SID>/control01.ctl",
                  "<ORACLE_MNT2>/oradata/<ORACLE_SID>/control02.ctl",
                  "<ORACLE_MNT2>/oradata/<ORACLE_SID>/control03.ctl" )

open_cursors = 100
max_enabled_roles = 40
db_block_buffers = 6144

shared_pool_size = 52428800

large_pool_size = 614400
java_pool_size = 20971520

log_checkpoint_interval = 10000
log_checkpoint_timeout = 1800

processes = 50

log_buffer = 163840

# audit_trail = false  # if you want auditing
# timed_statistics = false  # if you want timed statistics
# max_dump_file_size = 10000  # limit trace file size to 5M each

# Uncommenting the lines below will cause automatic archiving if archiving has
# been enabled using ALTER DATABASE ARCHIVELOG.
# log_archive_start = true
# log_archive_dest_1 = "location=<ORACLE_BASE>/admin/<ORACLE_SID>/arch"
# log_archive_format = arch_%t_%s.arc

# If using private rollback segments, place lines of the following
# form in each of your instance-specific init.ora files:
rollback_segments = ( RBS0, RBS1, RBS2, RBS3, BIG_RBS )

# Global Naming -- enforce that a dblink has same name as the db it connects to
# global_names = false

# Uncomment the following line if you wish to enable the Oracle Trace product
# to trace server activity.  This enables scheduling of server collections
# from the Oracle Enterprise Manager Console.
# Also, if the oracle_trace_collection_name parameter is non-null,
# every session will write to the named collection, as well as enabling you
# to schedule future collections from the console.
# oracle_trace_enable = true

# define directories to store trace and alert files
background_dump_dest = <ORACLE_BASE>/admin/<ORACLE_SID>/bdump
core_dump_dest = <ORACLE_BASE>/admin/<ORACLE_SID>/cdump
#Uncomment this parameter to enable resource management for your database.
#The SYSTEM_PLAN is provided by default with the database.
#Change the plan name if you have created your own resource plan.
# resource_manager_plan = system_plan
user_dump_dest = <ORACLE_BASE>/admin/<ORACLE_SID>/udump

db_block_size = 8192

remote_login_passwordfile = exclusive

os_authent_prefix = ""

# The following parameters are needed for the Advanced Replication Option
# Commented out because distributed processing is not being used
#job_queue_processes = 4
#job_queue_interval = 60
#distributed_transactions = 10
```

```
#open_links = 4

mts_dispatchers = "(PROTOCOL=TCP)(PRE=oracle.aurora.server.SGiopServer)"
# Uncomment the following line when your listener is configured for SSL
# (listener.ora and sqlnet.ora)
# mts_dispatchers = "(PROTOCOL=TCPS)(PRE=oracle.aurora.server.SGiopServer)"

compatible = "8.1.0"
sort_area_size = 65536
sort_area_retained_size = 65536

=======================================
== File data/dbdesc/create/SIDrun.sh ==
=======================================

#!/bin/sh

################################################
#
# Modified by K. Dritz, ANL, 25 Nov. 2000
# to increase size of redo*.log files to 5M
#
################################################

ORACLE_SID=<ORACLE_SID>
export ORACLE_SID

<ORACLE_HOME>/bin/svrmgrl << EOF
spool <ORACLE_BASE>/admin/<ORACLE_SID>/create/crdb1.log
connect internal
startup nomount pfile =
   "<ORACLE_BASE>/admin/<ORACLE_SID>/pfile/init<ORACLE_SID>.ora";
CREATE DATABASE "<ORACLE_SID>"
   MAXDATAFILES 254
   MAXINSTANCES 8
   MAXLOGFILES 32
   CHARACTER SET US7ASCII
   NATIONAL CHARACTER SET US7ASCII
   DATAFILE '<ORACLE_MNT2>/oradata/<ORACLE_SID>/system01.dbf' SIZE 75M
   AUTOEXTEND ON NEXT 640K
   LOGFILE
      '<ORACLE_MNT1>/oradata/<ORACLE_SID>/redo01.log' SIZE 5M,
      '<ORACLE_MNT2>/oradata/<ORACLE_SID>/redo02.log' SIZE 5M,
      '<ORACLE_MNT3>/oradata/<ORACLE_SID>/redo03.log' SIZE 5M,
      '<ORACLE_MNT4>/oradata/<ORACLE_SID>/redo04.log' SIZE 5M;
disconnect
spool off
exit


EOF

========================================
== File data/dbdesc/create/SIDrun1.sh ==
========================================

#!/bin/sh

#########################################################################
#
# Modified by K. Dritz, ANL
# to increase size of tools*.dbf files to 35M
# and to increase size of rbs*.dbf files to 35M
# and to increase size of temp01.dbf file to 250M
# and to create a BIGROLL tablespace (1 datafile of size 500M)
# and to reduce rollback segments to five (RBS0..RBS3 plus BIG_RBS)
#
#########################################################################

ORACLE_SID=<ORACLE_SID>
export ORACLE_SID

<ORACLE_HOME>/bin/svrmgrl << EOF
spool <ORACLE_BASE>/admin/<ORACLE_SID>/create/crdb2.log
```

```
connect internal
@<ORACLE_HOME>/rdbms/admin/catalog.sql;

REM ********** ALTER SYSTEM TABLESPACE *********
ALTER TABLESPACE SYSTEM
    MINIMUM EXTENT 64K
    DEFAULT STORAGE
        ( INITIAL 64K
          NEXT 64K
          MINEXTENTS 1
          MAXEXTENTS UNLIMITED
          PCTINCREASE 50 );

REM ********** TABLESPACE FOR Tools **********
CREATE TABLESPACE TOOLS
    DATAFILE
        '<ORACLE_MNT2>/oradata/<ORACLE_SID>/tools02.dbf' SIZE 35M,
        '<ORACLE_MNT3>/oradata/<ORACLE_SID>/tools03.dbf' SIZE 35M,
        '<ORACLE_MNT4>/oradata/<ORACLE_SID>/tools04.dbf' SIZE 35M
    REUSE
    AUTOEXTEND ON NEXT 320K
    MINIMUM EXTENT 32K
    DEFAULT STORAGE
        ( INITIAL 32K
          NEXT 32K
          MINEXTENTS 1
          MAXEXTENTS 4096
          PCTINCREASE 0 );

REM ********** TABLESPACE FOR ROLLBACK **********
CREATE TABLESPACE RBS
    DATAFILE
        '<ORACLE_MNT2>/oradata/<ORACLE_SID>/rbs02.dbf' SIZE 35M,
        '<ORACLE_MNT3>/oradata/<ORACLE_SID>/rbs03.dbf' SIZE 35M,
        '<ORACLE_MNT4>/oradata/<ORACLE_SID>/rbs04.dbf' SIZE 35M
    REUSE
    AUTOEXTEND ON NEXT 5120K
    MINIMUM EXTENT 512K
    DEFAULT STORAGE
        ( INITIAL 512K
          NEXT 512K
          MINEXTENTS 8
          MAXEXTENTS 4096 );
CREATE TABLESPACE BIGROLL
    DATAFILE
        '<ORACLE_MNT3>/oradata/<ORACLE_SID>/bigroll.dbf' SIZE 500M
    REUSE
    AUTOEXTEND ON NEXT 10M
    MINIMUM EXTENT 10M
    DEFAULT STORAGE
        ( INITIAL 10M
          NEXT 10M
          MINEXTENTS 2
          MAXEXTENTS 4096 );

REM ********** TABLESPACE FOR TEMPORARY **********
CREATE TABLESPACE TEMP
    DATAFILE '<ORACLE_MNT3>/oradata/<ORACLE_SID>/temp01.dbf' SIZE 250M
    REUSE
    AUTOEXTEND ON NEXT 640K
    MINIMUM EXTENT 64K
    DEFAULT STORAGE
        ( INITIAL 64K
          NEXT 64K
          MINEXTENTS 1
          MAXEXTENTS UNLIMITED
          PCTINCREASE 0)
    TEMPORARY;

REM ********** TABLESPACE FOR USER **********
CREATE TABLESPACE USERS
    DATAFILE '<ORACLE_MNT4>/oradata/<ORACLE_SID>/users01.dbf' SIZE 10M
    REUSE
    AUTOEXTEND ON NEXT 1280K
    MINIMUM EXTENT 128K
    DEFAULT STORAGE
        ( INITIAL 128K
          NEXT 128K
          MINEXTENTS 1
          MAXEXTENTS 4096
          PCTINCREASE 0 );

REM ********** TABLESPACE FOR INDEX **********
CREATE TABLESPACE INDX
    DATAFILE '<ORACLE_MNT3>/oradata/<ORACLE_SID>/indx01.dbf' SIZE 54M
    REUSE
    AUTOEXTEND ON NEXT 1280K
    MINIMUM EXTENT 128K
    DEFAULT STORAGE
        ( INITIAL 128K
          NEXT 128K
          MINEXTENTS 1
          MAXEXTENTS 4096
          PCTINCREASE 0 );

REM **** Creating four rollback segments ****************
REM **** Plus BIG_RBS **********************************
CREATE PUBLIC ROLLBACK SEGMENT RBS0
    TABLESPACE RBS
    STORAGE ( OPTIMAL 4096K );
CREATE PUBLIC ROLLBACK SEGMENT RBS1
    TABLESPACE RBS
    STORAGE ( OPTIMAL 4096K );
CREATE PUBLIC ROLLBACK SEGMENT RBS2
    TABLESPACE RBS
    STORAGE ( OPTIMAL 4096K );
CREATE PUBLIC ROLLBACK SEGMENT RBS3
    TABLESPACE RBS
    STORAGE ( OPTIMAL 4096K );
REM CREATE PUBLIC ROLLBACK SEGMENT RBS4 TABLESPACE RBS
REM STORAGE ( OPTIMAL 4096K );
REM CREATE PUBLIC ROLLBACK SEGMENT RBS5 TABLESPACE RBS
REM STORAGE ( OPTIMAL 4096K );
REM CREATE PUBLIC ROLLBACK SEGMENT RBS6 TABLESPACE RBS
REM STORAGE ( OPTIMAL 4096K );
CREATE PUBLIC ROLLBACK SEGMENT BIG_RBS
    TABLESPACE BIGROLL
    STORAGE ( OPTIMAL 40M );
ALTER ROLLBACK SEGMENT "RBS0" ONLINE;
ALTER ROLLBACK SEGMENT "RBS1" ONLINE;
ALTER ROLLBACK SEGMENT "RBS2" ONLINE;
ALTER ROLLBACK SEGMENT "RBS3" ONLINE;
REM ALTER ROLLBACK SEGMENT "RBS4" ONLINE;
REM ALTER ROLLBACK SEGMENT "RBS5" ONLINE;
REM ALTER ROLLBACK SEGMENT "RBS6" ONLINE;
ALTER ROLLBACK SEGMENT "BIG_RBS" ONLINE;


REM **** SYS and SYSTEM users ****************
alter user sys temporary tablespace TEMP;
alter user system temporary tablespace TEMP;
disconnect
spool off
exit


EOF

========================================
== File data/dbdesc/create/SIDrun2.sh ==
========================================

#!/bin/sh
ORACLE_SID=<ORACLE_SID>
export ORACLE_SID

<ORACLE_HOME>/bin/svrmgrl << EOF
spool <ORACLE_BASE>/admin/<ORACLE_SID>/create/crdb3.log
```

```
connect internal
@<ORACLE_HOME>/rdbms/admin/catproc.sql
@<ORACLE_HOME>/rdbms/admin/caths.sql
@<ORACLE_HOME>/rdbms/admin/otrcsvr.sql
connect system/manager
@<ORACLE_HOME>/sqlplus/admin/pupbld.sql

disconnect
spool off
exit


EOF

END OF FILE
```

A.5 PostInstall File

```
#!/bin/sh

# This is the PostInstall script for the ELIST Database Instance segment.

# Its purpose is to make sure that a database instance suitable for ELIST exists
# for later segments of the ELIST mission application to use.  When it runs,
# during installation, it allows the person doing the install (assumed to be a
# DBA) the choice of using an existing instance or of creating a new instance.

# The ORADBI segment can be assumed to exist because this segment requires it.
# Set ORADBI_HOME.

ORADBI_HOME=`COEFindSeg 'ORACLE DataBase Instance' | awk -F: '{print $1}'`
export ORADBI_HOME

# Fix the ownership of /h/COTS/ORADBI/data/ELISTdbinst.  (It is wrong, due to
# the non-implementation of the $ExternalData keyword of FileAttribs, and that
# prevents successful installation.)  The chown below requires root permission,
# but this script has to run as root anyway, for other reasons.

chown 200:102 $ORADBI_HOME/data/ELISTdbinst

# Get the SIDs (names) of existing database instances, if any.

SIDS=`$ORADBI_HOME/bin/ORADBI_getSids`

# If no instances exist, force the installer to create a new instance.
# Otherwise, give the installer a choice between using an existing instance
# for ELIST or creating a new one.

FOUND='The following database instances were found:'

if [ -z "$SIDS" ] ; then

   # No instances exist.

   NEWINST=yes
   SIDS_MSG=`echo "$FOUND"; echo '(none)'`

else

   # Instances exist.

   SIDS_MSG=`echo "$FOUND"; echo "$SIDS"`

   MSG=`echo "$SIDS_MSG"; \
        echo 'Do you want to use an existing instance or create a new one?'`
   COEAskUser -B 'Use existing instance' 'Create new instance' "$MSG"

   if [ $? -eq 1 ] ; then

      # Installer wants to use an existing instance.

      NEWINST=no
      # Display names of available instances, and prompt the installer until he
      # enters the name of an existing instance or asks to cancel the install.

      while [ true ] ; do

         MSG=`echo "$SIDS_MSG"; \
              echo 'Enter one of the above names, or type a null line to'; \
              echo 'cancel the install.  Note that you are responsible for'; \
              echo 'making sure that the instance you choose has the'; \
              echo 'required properties, a summary of which can be found'; \
              echo 'in the ReleaseNotes for this segment.'`
         MAXSID=`echo "$SIDS" | awk '{l=length; if (l>ml) ml=l} END {print ml}'`
         SID=`COEPrompt -C $MAXSID "$MSG" | awk 'NR==2 {print $1}' | \
              tr '[:lower:]' '[:upper:]'`

         if [ -z "$SID" ] ; then

            # Response was either an empty string or all blanks.
            # Installer wants to abort.  Tell the COE Installer that.

            COEInstError 'Aborting this install, as requested.' || exit

         else

            # Response is presumably an instance name.
            # Installer wants to use an existing instance.

            # Check that the instance really exists.
            # Break the loop if it does, otherwise repeat the loop.

            echo "$SIDS" | /bin/egrep -i -s -e "$SID" && break
            COEMsg "$SID is not the name of an existing instance!"

         fi # [ -z "$SID" ]

      done

   else

      # Installer wants to create a new instance.

      NEWINST=yes

   fi # [ $? -eq 1 ]

fi # [ -z "$SIDS" ]

# At this point, either NEWINST = no and SID is the name of an existing instance
# to use for ELIST, or NEWINST = yes.  In the latter case, prompt the installer
# until he enters the name of a non-existent instance or asks to cancel the
# install.  Instance names are limited to 8 characters in length (inherited from
# ORADBI, if not Oracle itself).

if [ $NEWINST = yes ] ; then

   while [ true ] ; do

      MSG=`echo "$SIDS_MSG"; \
           echo 'Enter a new instance name, or type a null line to'; \
           echo 'cancel the install.'`
      SID=`COEPrompt -C 8 "$MSG" | awk 'NR==2 {print $1}' | \
           tr '[:lower:]' '[:upper:]'`

      if [ -z "$SID" ] ; then

         # Response was either an empty string or all blanks.
         # Installer wants to abort.  Tell the COE Installer that.

         COEInstError 'Aborting this install, as requested.' || exit

      else

         # Response is presumably an instance name.
         # Check that the instance does not exist and that the name is short
         # enough to satisfy ORADBI_auto.  Break the loop if both checks
         # succeed, otherwise display a message and repeat the loop.

         echo "$SIDS" | /bin/egrep -i -s -e "$SID"
         if [ $? -eq 0 ] ; then
            # Instance exists.
            COEMsg "$SID already exists!"
            continue
         fi

         # The name is indeed that of a non-existent instance.

         break

      fi # [ -z "$SID" ]

   done

   # At this point, NEWINST = yes and SID is the name of a non-existent
   # instance to be created.

   # Register the database instance description to be linked into ORADBI's
   # directory of instance descriptions.  Note that the following invocation of
   # ORADBI_register requires this script to run as root (the $ROOT keyword
   # applies).

   $ORADBI_HOME/bin/ORADBI_register '' ELISTdbinst
   STATUS=$?

   # Process the return code.  If nonzero, display the appropriate messages and
   # abort the install.

   if [ $STATUS -ne 0 ] ; then

      MSG=`echo "The $SID database instance has not been created"; \
           echo 'because ORADBI_register failed to register the ELIST DB'; \
           echo "Description; it returned an error code of $STATUS."`

      case $STATUS in

      1)

         MSG=`echo "$MSG"; \
              echo 'That error code means that ORADBI_register failed to'; \
              echo 'link the ELIST DB Description into its data directory.'`

      ;; # 1)

      10)

         MSG=`echo "$MSG"; \
              echo 'That error code means that the ELIST DB Description has'; \
              echo 'already been registered.'`

      ;; # 10)

      esac # $STATUS

      MSG=`echo "$MSG"; \
           echo "Consult the System Administrator's Manual (SAM) for this"; \
           echo 'segment.  Aborting the install.'`

      COEInstError "$MSG" || exit

   fi # [ $STATUS -ne 0 ]

   # Invoke ORADBI_auto to:
   #   (a) prompt for the desired password for the SYSTEM user of the new
   #       database instance and then
   #   (b) create the instance.
   # (It actually creates the instance with the default password and then
   # changes it.  However, if a password is not successfully entered, it
   # returns exit status 12 and does not create an instance.  ORADBI_auto is
```

```
   # invoked in a loop until the exit status is other than 12.)  Note that the
   # following invocation of ORADBI_auto requires this script to run as root
   # (the $ROOT keyword applies).

   PASSWORD_ENTERED=no

   while [ $PASSWORD_ENTERED = no ] ; do
      $ORADBI_HOME/bin/ORADBI_auto '' \
         -passwd SYSTEM "$SID" "$SID.WORLD" `hostname` ELISTdbinst
      STATUS=$?
      if [ $STATUS -eq 12 ] ; then
         MSG=`echo 'You did not successfully enter a password to be'; \
              echo 'assigned to the SYSTEM account of the new database'; \
              echo 'instance.  Please try again.'`
         COEMsg "$MSG"
      else
         PASSWORD_ENTERED=yes
      fi
   done

   # Process the return code.  Display the appropriate messages.  Beyond this
   # point, the segment is left installed, regardless of any failures.
   # Deinstalling it should successfully clean up any vestiges of the
   # instance, however.

   if [ $STATUS -eq 0 -o $STATUS -eq 21 ] ; then
      MSG="The $SID database instance has been created."
      if [ $STATUS -eq 21 ] ; then
         MSG=`echo "$MSG"; \
              echo 'However, ORADBI_auto failed to change the SYSTEM'; \
              echo 'password to the value you supplied.'`
      fi
   else
      MSG=`echo "The $SID database instance has not been created, and"; \
           echo "ORADBI_auto returned an error code of $STATUS."`
      if [ $STATUS -eq 20 ] ; then
         MSG=`echo "$MSG"; \
              echo 'That error code means that Oracle failed to create'; \
              echo 'the instance.'`
      fi
      MSG=`echo "$MSG"; \
          echo 'Although the instance is not functional, the segment'; \
          echo 'will remain installed.'`
      LOG="$ORADBI_HOME/data/db/$SID.log"
      if [ -f "$LOG" ] ; then
        MSG=`echo "$MSG"; \
             echo 'This allows information on the failure to be gathered'; \
             echo "by studying the log file, $LOG."`
      fi
      MSG=`echo "$MSG"; \
          echo 'Deinstalling the segment should successfully remove any';  \
          echo 'vestiges of the instance.'`
   fi

   if [ $STATUS -ne 0 ] ; then
      MSG=`echo "$MSG"; \
          echo 'For further information, consult the System'; \
           echo "Administrator's Manual (SAM) for this segment."`
   fi

else

   MSG=`echo "Because the $SID database instance existed prior to the"; \
        echo 'installation of this segment, it will remain after the segment'; \
        echo 'is deinstalled in the future.'`

fi # [ $NEWINST = yes ]

COEMsg "$MSG"

# At this point, SID is the name of the ELIST instance, and NEWINST = yes if it
# was created by this segment.

# Save data in the Instance_Info file identifying the instance to other segments
# and noting for the DEINSTALL script that this segment created it.

# Note that this segment is the only one that writes to the Instance_Info file
# in the ELIST Global Data Segment.  The FileAttribs descriptor of that segment
# makes Instance_Info read-only, but this segment can write to it because it
# runs as root (the $ROOT keyword applies), for the reasons documented in this
# segment's IntgNotes file.  The use of root privileges for this purpose is not
# essential, however, and other approaches could have been implemented.

# The following reference to ELISTglob/data is valid, because the ELIST Global
# Data Segment is required by this segment and must therefore already have been
# installed.

INFO=$DATA_DIR/global/ELISTglob/data/Instance_Info

# Before writing lines 3-6 into the file, reset it to the initial state of
# two lines.  This is done in case this segment was previously installed and
# was deinstalled.  (It is not reset to the initial state at that time, in
# case the file is found to be corrupted; in that case, preserving it allows
# the problem to be investigated after the deinstallation.)

HEADER1=\
'This file is filled in during the installation of the ELIST Database Instance'
HEADER2=\
'Segment.  The information is used by other ELIST mission application segments.'
echo "$HEADER1" > $INFO
echo "$HEADER2" >> $INFO

echo "ELIST instance name=$SID" >> $INFO
echo "Created by ELIST Database Instance Segment=$NEWINST" >> $INFO
echo "On server `hostname`" >> $INFO
echo "On date `date`" >> $INFO

# Also empty the lists of users with the elist and etpfdd roles at the time
# of the last deinstall of the ELIST Database Segment.  These lists should
# logically be emptied when THIS segment is deinstalled, but for uniformity
# that action is deferred until its reinstallation.

# Although these lists are owned by elistown and have write permission only
# for the user, this PostInstall runs as root and root can write into any
# files.

DB_USERS=$DATA_DIR/global/ELISTglob/data/Users_When_ELISTdb_Deinstalled
for role in elist etpfdd ; do
   cat /dev/null > $DB_USERS/Users_With_The_${role}_user_Role
   # /dev/null behaves like an empty file
done

exit 0

# END OF FILE
```

A.6 DEINSTALL File

```
#!/bin/sh

# This is the DEINSTALL script for the ELIST Database Instance segment.

# Check whether segments that depend on this one have already been deinstalled.
# If not, display a message and abort; the installer should deinstall those
# segments first.  This ensures that they do not remain on the system without
# the segments on which they depend, which would cause problems if one tried
# to "use" them.  Aborting this deinstall is tolerable here, because there is
# a clear way to make progress.

# Check the ELIST Database Fill Segment.  If found, abort as described above.

if [ "`COEFindSeg 'ELIST Database Fill Segment'`" != 'Not Found' ] ; then

   MSG=`echo 'The ELIST Database Fill Segment, which depends on the segment'; \
        echo 'you are trying to deinstall, is still installed.  You must'; \
        echo 'deinstall it before you can deinstall this segment.  Aborting'; \
        echo 'the deinstall.'`
   COEInstError "$MSG" || exit

fi

# The ELIST Database Fill Segment is unusual in that it is quite normal for it
# to be deinstalled, to save space, after the successful installation of the
# ELIST Database Segment.  Thus, the ELIST Database Fill Segment's DEINSTALL
# script does NOT check for, and complain about, the presence of other ELIST
# segments installed after it and not yet deinstalled.  Therefore, it is
# necessary to check for the first of those here.  It, in turn, checks for the
# next, and so on.

# Check the ELIST Database Segment.  If found, abort as described above.

if [ "`COEFindSeg 'ELIST Database Segment'`" != 'Not Found' ] ; then

   MSG=`echo 'The ELIST Database Segment, which depends indirectly on the'; \
        echo 'segment that you are trying to deinstall, is still installed.'; \
        echo 'You must deinstall it before you can deinstall this segment.'; \
        echo 'Aborting the current deinstall.'`
   COEInstError "$MSG" || exit

fi

# Beyond this point, this segment will definitely be deinstalled (i.e., the
# current deinstall will not be deliberately aborted by calling COEInstError),
# regardless of what failure might be detected.  This ensures that it is always
# possible to deinstall this segment, even if it means leaving some of the
# external effects of its installation unreversed.  The installer is directed
# to the SAM for this segment, which gives tips that a DBA might find useful in
# recovering, by means outside this segment, from the failure to reverse the
# external effects of its installation.  (Actually, it is likely that those
# effects WILL be reversed successfully.)

# Extract information from the Instance_Info file in the ELIST Global Data
# Segment and use it to determine what instance, if any, to delete.  If the
# information appears to be missing, the file may have been corrupted.  In that
# case, display a message and proceed with the removal of this segment without
# using the facilities of ORADBI to delete any database instance.

# The ELIST Global Data Segment can be assumed to exist, because it can't be
# deinstalled if this segment still exists.  Hence the following reference is
# valid.

INFO=$DATA_DIR/global/ELISTglob/data/Instance_Info

SID=`awk -F= 'NR==3 {print $2}' "$INFO"`
NEWINST=`awk -F= 'NR==4 {print $2}' "$INFO"`
if [ -z "$SID" -o -z "$NEWINST" ] ; then
   NEWINST=unknown
fi

case $NEWINST in

unknown)

   # The Instance_Info file is missing, or the information in it is not intact.

   MSG=`echo 'The Instance_Info file in the ELIST Global Data Segment, which'; \
        echo 'was written when this segment was installed, is missing,'; \
        echo 'incomplete, or garbled.  The name of the database instance'; \
        echo 'used for ELIST is unavailable, and furthermore it is unknown'; \
        echo 'whether that instance existed prior to the installation of'; \
        echo 'this segment or was created by its installation.  The removal'; \
        echo 'of this segment will proceed, but without doing anything to'; \
        echo 'delete any database instance.  If there really is an instance'; \
        echo 'to be deleted, a DBA will have to delete it by methods'; \
        echo "outlined in the System Administrator's Manual (SAM) for this"; \
        echo 'segment.'`
   COEMsg "$MSG"

;; # unknown)

yes)
```

```
# The information in the Instance_Info file appears to be intact, and
# this segment created the instance (during its installation).

# Check whether ORADBI is installed.  That segment must have been installed
# to have installed this segment, but it could have been subsequently
# removed.  The reversal of the external effects of installing this segment
# requires the use of facilities of ORADBI.  If it is not now present, there
# is not much that can be done here.  It does not make a lot of sense to
# abort this deinstall, and insist on the reinstallation of ORADBI before
# proceeding, because the ELIST instance may have been wiped away when
# ORADBI was removed; at least, the registration information that it retains
# is gone and cannot be reconstituted.

ORADBI_HOME=`COEFindSeg 'ORACLE DataBase Instance' | awk -F: '{print $1}'`

if [ "$ORADBI_HOME" = 'Not Found' ] ; then

   MSG=`echo 'The ORACLE DataBase Instance (ORADBI) segment, which is'; \
        echo 'required, has been removed, hence there is no way for this'; \
        echo "DEINSTALL script to delete the $SID instance.  The removal"; \
        echo 'of this segment will proceed, but without doing anything to'; \
        echo "delete the $SID instance.  A DBA will have to delete it by"; \
        echo "methods outlined in the System Administrator's Manual (SAM)"; \
        echo 'for this segment.'`
   COEMsg "$MSG"

else

   # ORADBI is still installed.  ORADBI_HOME is usable.

   export ORADBI_HOME

   # Invoke ORADBI_auto to delete the instance.

   $ORADBI_HOME/bin/ORADBI_auto '' "$SID" -r
   STATUS=$?

   case $STATUS in

   0)

      MSG="The $SID database instance has been successfully removed."

   ;; # 0)

   *)

      MSG=`echo "ORADBI_auto returned an error code of $STATUS during"; \
           echo 'the removal of the database instance.'`

      case $STATUS in

      30)

         MSG=`echo "$MSG"; \
              echo 'That error code means that ORADBI_auto failed to'; \
              echo 'remove the instance.'`
         LOG="$ORADBI_HOME/data/db/$SID.log"
         if [ -f "$LOG" ] ; then
            MSG=`echo "$MSG"; \
                 echo 'Additional information may be found in the log'; \
                 echo "file, $LOG."`
         fi

      ;; # 30)

      esac # $STATUS

      MSG=`echo "$MSG"; \
           echo 'Despite the error, the deinstallation of this segment'; \
           echo 'will continue, and any vestiges of the instance will'; \
           echo 'probably be cleaned up successfully.  For further'; \
           echo "information, consult the System Administrator's"; \
              echo 'Manual (SAM) for this segment.'`

      ;; # *)

      esac # $STATUS

      COEMsg "$MSG"

      # Unregister the database instance description previously registered by
      # PostInstall.

      $ORADBI_HOME/bin/ORADBI_register '' -r ELISTdbinst
      STATUS=$?

      if [ $STATUS -ne 0 ] ; then

         MSG=`echo 'ORADBI_register failed to unregister the ELIST DB'; \
              echo "Description (it returned an error code of $STATUS)."`

         case $STATUS in

         1)

            MSG=`echo "$MSG"; \
                 echo 'That error code means that ORADBI_register failed'; \
                 echo 'to unlink the ELIST DB Description.'`

         ;; # 1)

         esac # $STATUS

         MSG=`echo "$MSG"; \
              echo 'Despite the error, the deinstallation of this segment'; \
              echo 'will continue.  For further information, consult the'; \
              echo "System Administrator's Manual (SAM) for this segment."`

         COEMsg "$MSG"

      fi # [$STATUS -ne 0 ]

   fi # [ "$ORADBI_INFO" = 'Not Found' ]

;; # yes)

no)

   MSG=`echo "The $SID database instance has not been removed"; \
        echo 'because it existed prior to the installation of this segment.'`
   COEMsg "$MSG"

;; # no)

esac # $NEWINST

# Note that the Instance_Info file should be restored at this point to the
# state it had when this segment was installed, i.e., just the first two
# header lines.  However, it is left intact, in case it was found to be
# corrupted above and needs to be examined later.  Instead, when this segment
# is installed, it resets the file to the starting state before writing into
# it.  Thus, a subsequent reinstallation of this segment will not accumulate
# additional lines into this file, despite its not being reset now.

# By the same token, the lists of DB users with the elist and etpfdd roles
# at the time of the last deinstallation of the ELIST Database Segment
# should now be restored to the state they had when this segment was installed,
# i.e., they should be emptied.  For uniformity, they are left intact, and
# instead they are reset to empty lists when this segment is reinstalled.

exit 0

# END OF FILE
```

A.7 FileAttribs File

```
# This is the FileAttribs descriptor for the ELIST Database Instance Segment.

$SegDir:750:418:102
$ExternalData:750:200:102
750:418:419:SegDescrip
750:418:419:SegDescrip/DEINSTALL
640:418:419:SegDescrip/FileAttribs
750:418:419:SegDescrip/PostInstall
640:418:419:SegDescrip/ReleaseNotes
640:418:419:SegDescrip/SegInfo
640:418:419:SegDescrip/SegName
640:418:419:SegDescrip/VERSION
640:418:419:SegDescrip/Validated
750:200:102:data
750:200:102:data/dbdesc
640:200:102:data/dbdesc/SID.cfg
750:200:102:data/dbdesc/create
750:200:102:data/dbdesc/create/Create_SID
750:200:102:data/dbdesc/create/SIDalterTablespace.sh
750:200:102:data/dbdesc/create/SIDjava.sh
750:200:102:data/dbdesc/create/SIDrun.sh
750:200:102:data/dbdesc/create/SIDrun1.sh
750:200:102:data/dbdesc/create/SIDrun2.sh
750:200:102:data/dbdesc/create/SIDsqlplus.sh
640:200:102:data/dbdesc/create/listener.SID
640:200:102:data/dbdesc/create/tnsnames.SID
750:200:102:data/dbdesc/pfile
640:200:102:data/dbdesc/pfile/orapwSID
640:200:102:data/dbdesc/pfile/initSID.ora
750:200:102:data/dbdesc/adhoc
750:200:102:data/dbdesc/arch
750:200:102:data/dbdesc/bdump
750:200:102:data/dbdesc/cdump
750:200:102:data/dbdesc/exp
750:200:102:data/dbdesc/udump
750:418:419:Integ
640:418:419:Integ/IntgNotes
640:418:419:Integ/VSOutput

# END OF FILE
```

Appendix B. Contents of the Files in the `Integ` Directory

The contents of the files in the `Integ` directory are reproduced in this appendix.

B.1 `IntgNotes` File

```
This is the IntgNotes file for the ELIST Database Instance Segment.

This submission Version 8.1.0.0 is the initial submission of the segment.

The ELIST Database Instance Segment is part of the ELIST mission application.
There are six other segments.  All seven segments have the same segment prefix
(ELIST).

The installation of this segment creates the ELIST database instance, and ints
deinstallation removes the instance.  However, the administrator is given the
option of using an already existing instance (which is not removed when the
segment is deinstalled).

Refer to the "Introduction to the Enhanced Logistics Intratheater Support Tool
(ELIST) Mission Application and its Segments: Global Data Segment, Database
Instance Segment, Database Fill Segment, Database Segment, Database Utility
Segment, Software Segment, and Reference Data Segment" for the following:

   + an overview of the mission application and all of its segments in the
     context of the application;

   + the definitions of key concepts and terms used throughout the ELIST
     documentation;

   + a complete list of the available ELIST documentation;

   + a brief history of ELIST; and

   + basic information pertinent to the client/server configuration and
     installation of the ELIST segments.

END OF FILE
```

B.2 Annotated `VSOutput` File

```
(W)-------------------------------------------------------------|
  Variable ORADBI_HOME must begin with the segment prefix (ELIST_)
(W)-------------------------------------------------------------|
  Variable ORADBI_HOME must begin with the segment prefix (ELIST_)

*** Explanation:

This segment invokes public APIs of the ORADBI segment in accordance with
guidance to developers documented in the SAM for ORADBI.  Those APIs expect
ORADBI_HOME to be already set; except for that, they set up the rest of their
environment by themselves.

This segment therefore sets ORADBI_HOME by processing the output of COEFindSeg,
which is used to locate ORADBI.  ORADBI_HOME has to be exported in order to be
used by the scripts (namely, the APIs).  This is done in PostInstall and in
DEINSTALL, accounting for the two instances of the above warning.


  Results of verification:
  "./ELISTdbinst":
       Totals
       ------
       Errors:    0
       Warnings:  2
```

This page intentionally left blank.

Appendix C. Chief Engineer Permissions and Waivers Requested

Chief Engineer Permission Request

SEGMENT DESIGNATION:	ELIST Database Instance Segment (ELIST)
CM Number:	10102
OPERATING SYSTEM:	Solaris 7
MISSION APPLICATION NAME:	Enhanced Logistics Intratheater Support Tool
TECHNICAL POC:	Kenneth W. Dritz Decision and Information Sciences Division Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439 dritz@anl.gov
APPROVAL REQUESTED FROM:	GCSS Chief Engineer
CHECKLIST ITEM:	5-7, Security Services section, "(UNIX) If privileged user permissions are required during segment installation or removal, the Chief Engineer has granted prior approval."
APPROVAL REQUESTED:	To use the `$ROOT` keyword for this segment's `PostInstall` and `DEINSTALL` scripts.
RATIONALE:	This is necessary for two reasons, the first permanent and the second transitory: (1) The `PostInstall` and `DEINSTALL` scripts of this segment both use the `ORADBI_register` and `ORADBI_auto` APIs of the ORACLE DataBase Instance (ORADBI) segment to create (in `PostInstall`) and delete (in `DEINSTALL`) a custom database instance in accord with the instructions in the SAM for the Oracle COTS products (ORADBI and others). Those instructions *require* the scripts executing these APIs to run as `root`, because the APIs `su` to a file owner account, and `root` permission is needed to avoid a password prompt on that `su`. (2) See the Rationale for the Permission Request associated with checklist item 6-53 for this segment.

Chief Engineer Permission Request

SEGMENT DESIGNATION: ELIST Database Instance Segment (ELIST)

CM Number: 10102

OPERATING SYSTEM: Solaris 7

MISSION APPLICATION NAME: Enhanced Logistics Intratheater Support Tool

TECHNICAL POC:
Kenneth W. Dritz
Decision and Information Sciences Division
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
dritz@anl.gov

APPROVAL REQUESTED FROM: GCSS Chief Engineer

CHECKLIST ITEM: 6-5, Security Services section, “No directory or file permission, whether created at install time or runtime, is less restrictive than identified in the Security chapter’s directory/file permissions table, unless approved by the Chief Engineer. Any such directories or files that do not meet the permissions identified in the table are documented in the *SVD* document or its equivalent. `VerifySeg` also uses this table to check permissions. All security-related messages from `VerifySeg` are explained in the `VSOutput` file and documented in the *SVD* document or its equivalent.”

APPROVAL REQUESTED: To give execute permission to some of the files underneath this segment’s `data` directory.

RATIONALE: Execute permission is not permitted by I&RTS Table 4-1 for files underneath a segment’s `data` directory. However, the `data` directory of this segment has a content that is prescribed by the ORACLE DataBase Instance (ORADBI) segment, which this segment uses, and that content involves, in part, files that must be executable. They are scripts (with the `.sh` extension) that Oracle executes (via the `ORADBI_auto` API in ORADBI) to create the ELIST database instance. These scripts are effectively part of the custom database description. We have absolutely no control over this; we are not at liberty to put these scripts anywhere else. The directory structure for the database description is represented by the `dbdesc` subdirectory of the `data` directory, and it has the organization (and file permissions and ownerships) required by ORADBI. ORADBI itself contains several “canned” database descriptions organized in exactly the same way, with executable scripts included. Note that `VerifySeg` does *not* complain about the execute permission, so no annotation is required in `VSOutput`.

Chief Engineer Permission Request

SEGMENT DESIGNATION:	ELIST Database Instance Segment (ELIST)
CM Number:	10102
OPERATING SYSTEM:	Solaris 7
MISSION APPLICATION NAME:	Enhanced Logistics Intratheater Support Tool
TECHNICAL POC:	Kenneth W. Dritz Decision and Information Sciences Division Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439 dritz@anl.gov
APPROVAL REQUESTED FROM:	GCSS Chief Engineer
CHECKLIST ITEM:	6-8, Standards Compliance section, “The segment is available on all COE-supported platforms unless otherwise approved by the Chief Engineer.”
APPROVAL REQUESTED:	To provide this segment only for the Sun/Solaris 7 and PC/NT (or Win2K) platforms. At the present time, only the Solaris 7 version has been registered and implemented, but the Windows version will be registered and implemented in the near future.
RATIONALE:	The sponsor, MTMC-TEA, only requires the ELIST mission application to operate on these platforms.

Chief Engineer Permission Request

SEGMENT DESIGNATION: ELIST Database Instance Segment (ELIST)

CM Number: 10102

OPERATING SYSTEM: Solaris 7

MISSION APPLICATION NAME: Enhanced Logistics Intratheater Support Tool

TECHNICAL POC:
Kenneth W. Dritz
Decision and Information Sciences Division
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
dritz@anl.gov

APPROVAL REQUESTED FROM: GCSS Chief Engineer

CHECKLIST ITEM: 6-53, Segment Descriptors section, "If any files need special permission/ownership settings, they are established through the `FileAttribs` descriptor if the descriptor supports the required setting. Exceptions to this are documented and approved by the Chief Engineer."

APPROVAL REQUESTED: To use a mechanism other than `FileAttribs` to establish an ownership setting, because `FileAttribs` does not support setting the ownership of a file outside the segment. The setting is established in `PostInstall`.

RATIONALE: The `/h/COTS/ORADBI/data/ELISTglob` directory (which is created by the Segment Installer because the ELIST Database Instance Segment is a data segment of segment scope, with ORADBI as the "affected segment," and which is given an ownership of `root:other` by the Segment Installer) must have an ownership of `oracle:dba` (`200:102`, as registered by the ORACLE RDBMS (ORAS) segment). This ownership must match that of the other directories and files in ORADBI, and it is needed for the APIs of ORADBI to access the custom database description in the ELIST Database Instance Segment via a symbolic link that the Segment Installer creates. (In the terminology of the SAM mentioned above in Permission Request 1 for this segment, the ELIST Database Instance Segment is an "ORADBI data segment.") Since the directory is external to the ELIST Database Instance Segment, the only way to use the `FileAttribs` descriptor to establish the required ownership is to use the `$ExternalData` keyword in that descriptor, which presumably affects this particular directory. However, the Segment Installer in 4.2.0.0P4 does not implement the `$ExternalData` keyword. The setting must therefore be established in `PostInstall`; furthermore, `PostInstall` must run as `root` because the UID under which it would otherwise run is not that of the directory whose ownership it needs to change. When the `$ExternalData` keyword is implemented, it will be possible to drop this Permission Request.

I&RTS Compliance Waiver Request

SEGMENT DESIGNATION:	ELIST Database Instance Segment (ELIST)
CM Number:	10102
OPERATING SYSTEM:	Solaris 7
MISSION APPLICATION NAME:	Enhanced Logistics Intratheater Support Tool
TECHNICAL POC:	Kenneth W. Dritz Decision and Information Sciences Division Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439 dritz@anl.gov
CHECKLIST ITEM:	6-52, Segment Descriptors section, "The `ReleaseNotes` descriptor conforms to the requirements stipulated in the Runtime Environment chapter and the Windows New Technology (NT)-Based Segments chapter."
RATIONALE:	The `ReleaseNotes` descriptor can perhaps be considered to violate this requirement because of its length. In order to be "meaningful to the System Administrator responsible for installing segments" [I&RTS Ch. 6], who must decide between using an existing database instance and letting `PostInstall` create a new database instance, we documented the minimal requirements that an existing instance must satisfy by copying into the `ReleaseNotes` descriptor certain database instance description files and creation scripts from the segment's `data` directory. This accounts for the length of the descriptor, and it coincidentally incorporates (via those copied database instance description files) the name of the individual and company responsible for developing them.
MITIGATION STRATEGY:	In the next release, we will substitute a concise statement of the requirements that must be satisfied and will conform in letter and in spirit to the requirement.

Documentation Waiver Request

SEGMENT DESIGNATION:	ELIST Database Instance Segment (ELIST)
CM Number:	10102
OPERATING SYSTEM:	Solaris 7
MISSION APPLICATION NAME:	Enhanced Logistics Intratheater Support Tool
TECHNICAL POC:	Kenneth W. Dritz Decision and Information Sciences Division Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439 dritz@anl.gov
DOCUMENTATION AFFECTED:	Installation Procedures (IP)
REQUEST:	To combine the Installation Procedures (IP) document for this segment with those of the other ELIST segments.
RATIONALE:	The installation and deinstallation of this segment are intimately related to those of the other six segments of the ELIST mission application, and all seven segments must be installed to use the features of ELIST. Certain steps must be taken, and certain choices considered, to prepare for the installation of these segments, and certain special requirements come into play between the installation of individual segments or when segments are deinstalled and replaced with later versions. For all these reasons, as well as to allow installation guidelines common to all the segments to be expressed in one place and to allow for helpful cross-referencing between the installation (or deinstallation) instructions of one segment and those of another, the roles and functions of the IP documents normally associated with the seven segments individually have been combined into a single IP document. All the information that would have been provided in seven separate IP documents is available in this one document. There are clearly identified sections for each segment. This organization of the material is believed to be uncommonly helpful to the end user (the installer).
MITIGATION STRATEGY:	Because much would be lost by conforming rigidly to the one-IP-per-segment expectation, there are no plans to do so.

Documentation Waiver Request

SEGMENT DESIGNATION:	ELIST Database Instance Segment (ELIST)
CM Number:	10102
OPERATING SYSTEM:	Solaris 7
MISSION APPLICATION NAME:	Enhanced Logistics Intratheater Support Tool
TECHNICAL POC:	Kenneth W. Dritz Decision and Information Sciences Division Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439 dritz@anl.gov
DOCUMENTATION AFFECTED:	Software Test Plan (STP), Software Test Description (STD), and Software Test Report (STR)
REQUEST:	To combine the Software Test Plan (STP), Software Test Description (STD), and Software Test Report (STR) documents for this segment into a single combined Software Test Plan/Description/Report (STP/STD/STR) document, and furthermore to combine the STP/STD/STR document for this segment with those of the other ELIST segments.
RATIONALE:	Several of the ELIST segments are passive (data) segments for which the only appropriate test is the verification that the segment appears in the Segment Installer's list of currently installed segments after installation. The installation and deinstallation testing of other segments involves verification that `PostInstall` and `DEINSTALL` behave as anticipated when interacting with the system administrator. Functional testing of the two software segments benefits from a carefully orchestrated order of testing, so that verified features can be used to verify other features. For all these reasons, and because there is really only one Software Test Plan for the entire ELIST mission application, it makes eminent sense to combine several manuals as described in the Request. All the information that would have been presented in separate documents is available in the combined document, with a minimum of repetition. This organization recognizes the essential unity of the ELIST mission application and the uselessness of individual ELIST segments except in the context of the mission application as a whole.
MITIGATION STRATEGY:	It is not practical or useful to conform to the rigid document guidelines concerning the STP/STD/STR documents, and there are no plans to do so.

This page intentionally left blank.